

Refine Search

Search Results -

Terms	Documents
L3 and sonet	1

Database:

US Pre-Grant Publication Full-Text Database
 US Patents Full-Text Database
 US OCR Full-Text Database
 EPO Abstracts Database
 JPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Search:

L16

Refine Search

Recall Text

Clear

Interrupt

Search History

DATE: Tuesday, November 16, 2004 [Printable Copy](#) [Create Case](#)

<u>Set Name</u> side by side	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u> result set
<i>DB=USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>			
<u>L16</u>	L3 and sonet	1	<u>L16</u>
<u>L15</u>	L13 and sonet	35	<u>L15</u>
<u>L14</u>	L13 ans sonet	216462	<u>L14</u>
<u>L13</u>	L9 and interleav\$	41	<u>L13</u>
<u>L12</u>	L9 and l3	0	<u>L12</u>
<u>L11</u>	L9 same fec	7	<u>L11</u>
<u>L10</u>	L9 same interleav\$	5	<u>L10</u>
<u>L9</u>	oc-48 same oc-192	220	<u>L9</u>
<u>L8</u>	oc-48 and oc-192	240	<u>L8</u>
<u>L7</u>	L6 same l1	5	<u>L7</u>
<u>L6</u>	(code adj1 word) or codeword	19364	<u>L6</u>
<u>L5</u>	L3 same iterat\$	4	<u>L5</u>
<u>L4</u>	L3 same permut\$	0	<u>L4</u>
<u>L3</u>	L2 same l1	189	<u>L3</u>

L2 (code ajd1 word) or codeword
L1 error near3 (inset\$ or inject\$)

2864562 L2
2180 L1

END OF SEARCH HISTORY

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

Generate Collection

Print

L7: Entry 1 of 5

File: USPT

Sep 16, 2003

DOCUMENT-IDENTIFIER: US 6622268 B2

**** See image for Certificate of Correction ****

TITLE: Method and apparatus for propagating error status over an ECC protected channel

Detailed Description Text (7):

FIG. 1 shows a block diagram of an example system for propagating error status over an ECC protected channel according to an example embodiment of the present invention. A first device 10 sends one or more code words, that includes data and check bits, across an ECC protected channel or interface 20 to a second device 30. First device 10 includes a controller 12, a code word generator 16, and an error injector circuit 14. Other devices and/or applications may also be present at first device 10 and still be within the spirit and scope of the present invention. A second device 30 includes a syndrome processor 32 and an error classifier circuit 34. Similarly, other devices may be present at second device 30 and still be within the spirit and scope of the present invention. Device 10 receives data at code word generator 16, and receives error status associated with the data at controller 12.

Detailed Description Text (10):

The data coming into code word generator 16 may consist of data words of any of many lengths, e.g., 16 bit data word, 32 bit data word, 64 bit data word, etc. Code word generator 16 receives the data and creates check bits for each data word. Controller 12 receives the error status and if the error status indicates an uncorrectable error, controller 12 controls error injector circuitry 14 to inject a triple-bit error into a four bit nibble of the code word before the code word is transmitted across channel 20. Syndrome processor 32 receives the code word with the triple bit error inserted in a nibble of the code word, and is therefore able to determine that the data being received has an uncorrectable error. Further, if a code word is sent with a triple-bit error inserted into a nibble of the code word, and channel 20 also injects an additional single-bit error into the code word (therefore producing four errors in the code word), the syndrome processor is still able to detect that the receive data has an uncorrectable error.

Detailed Description Text (15):

A check bit is generated for each row of the matrix (i.e., a data word). Eight check bits will be used also for this purpose. The example ECC code shown above is a (72,64) SEC-DED-S4D code, i.e. The code length is 72 bits, the data length is 64 bits, and there are 8 check bits. Thus, since there are eight check bits in this example ECC code, there are eight rows in the ECC code. The check bit columns are denoted by a "c" at the bottom of the column, the other positions are data. A "1" in a check bit column denotes the check bit position for the particular row that the "1" resides in. For example, looking at the first check bit column, note that a "1" exists only in row three of this check bit column. This denotes that the check bit which is calculated on the data of row three is to be placed in this position. The number of check bits are selected based on a desired level of error detection and correction. This ECC code provides the property that triple errors within a nibble of a code word received plus any additional single error (i.e., injected by the channel) are detectable by a receiving side (e.g., device 30).

Detailed Description Text (21):

FIG. 2 shows a flowchart of an example process for propagating error status over an ECC protected channel according to an example embodiment of the present invention. Device A receives data and an associated error status S1. Device A then uses an ECC code to generate check bits on the data S2. Device A generates a code word for each data word in the data received S3. The code word includes the data word and the generated check bits. A determination is made by device A as to whether the error status indicates that the received data has an uncorrectable error S4. If the error status indicates no error, or correctable error (indicating that the error has already been corrected in the data), the code word that has been generated is left untouched S5. However, if the error status indicates that the received data has an uncorrectable error, device A injects a triple-bit error into a nibble of the code word S6.

Detailed Description Text (22):

Device A sends the code word across the channel to device B, S7. By injecting a triple-bit error into a nibble of the code word, device A is transmitting the fact that the data has an uncorrectable error to the receiving device. A single-bit error may be injected into the code word by the channel due to noise on the channel or other factors S8. If the channel has not injected any errors into the code word, the code word will remain with three errors, if device A determined that the data had an uncorrectable error S9. If the data has an uncorrectable error, the code word will have a triple-bit error inserted in it. If the channel further injects an additional single bit error, the code word will now contain four errors S10.

Detailed Description Text (23):

Device B receives the code word and regenerates the check bits on the data using the ECC code S11. Device B computes a syndrome and uses the syndrome to classify any errors detected S12. By using the ECC code at device B, both triple-bit error in a nibble of the code word, and a triple-bit error in the nibble of the code word as well as a single bit error caused by the channel in the code word will both be classified as uncorrectable errors S13. Therefore, device B will have been made aware of an uncorrectable error in the data and will handle the data accordingly S14. A single error injected by the channel or any other source may occur anywhere in the code word and still be detected based on the ECC code.

CLAIMS:

8. The system according to claim 1, wherein the first device includes an error injection circuit, the error injection circuit injecting a triple error into a nibble of the at least one codeword if the error status indicated an uncorrectable error.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

Generate Collection

Print

L7: Entry 2 of 5

File: USPT

Sep 9, 2003

DOCUMENT-IDENTIFIER: US 6617985 B1

TITLE: Method and/or apparatus for implementing constraint codes with low error propagation

Detailed Description Text (21):

Referring to FIG. 7, an example of error propagation analysis 400 is shown. Error propagation analysis is often very complex and generally involves the statistical rate of occurrence of various errors and code words. To generate the plots shown in FIG. 7, errors may be injected into coded words. The number of ECC symbols (e.g., 10 bit symbols) that the were affected after decoding are generally evaluated. Errors may be injected with a statistical profile from a read channel simulation. For instance, an error was injected as a single bit 50% of the time, two bits 10% of the time, 3 bits 20% of the time, etc. Note that the error rate profile may be almost the same for the coded data (Post-RLL) as for the uncoded data (Pre-RLL).

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

Generate Collection

Print

L7: Entry 3 of 5

File: USPT

Mar 25, 2003

DOCUMENT-IDENTIFIER: US 6539503 B1

**** See image for Certificate of Correction ****

TITLE: Method and apparatus for testing error detection

Detailed Description Text (11):

A triggering condition or event that prompts the error injector 630 to dynamically inject an error into a codeword variable within the processor model 405' is a cache hit. As the error injector 630 monitors cache activity in the processor model 405', the error injector 630 intercepts reads to the cache model 450' and writes an erroneous version of the codeword into the cache model 450' before the cache read is performed. For example, the error injector 630 inverts one bit of a codeword, if a single parity coding scheme is utilized. In continued simulation of the operation of the processor model 405', the error detection/correction section model 475' detects the error soon thereafter, if the error detection/correction section model 475' is operating properly. By injecting errors when a cache hit has been detected in this way, the invention easily and precisely exercises the error detection/correction section model 475'. That is to say, the invention has a hit rate that is very high (if not certain). This in turn allows focused testing at any time during simulated operation, regardless of initialization conditions.

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

Generate Collection

Print

L10: Entry 2 of 5

File: USPT

Jan 27, 2004

DOCUMENT-IDENTIFIER: US 6684350 B1

TITLE: Repetitive pattern testing circuit for AC-coupled systems

Detailed Description Text (23):

FIG. 6 illustrates a block diagram of portions of protocol processor 520 that includes a framer 600. Deserializer 650 converts a serial incoming SONET signal 610 (e.g., an OC-192 data stream) from line side optical receivers (not shown) into a parallel bitstream 660 which is received by receive module 605. Receive module 605 optionally processes the forward error correction (FEC) information and de-interleaves the OC-192 signal into four OC-48 line rate signals 615 for delivery to downstream OC-48 processors. Transmit module 620 processes four incoming OC-48 system rate signals from the OC-48 processors (signals 625), optionally inserts forward error correction information, and interleaves the four OC-48 signals into an OC-192 signal 630 for transmission by line side optical transmitters (not shown). A CPU Interface module 635 provides the CPU connection to the internal device registers.

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

Generate Collection

Print

L10: Entry 3 of 5

File: USPT

Jun 10, 2003

DOCUMENT-IDENTIFIER: US 6577635 B2

TITLE: Data packet transmission scheduling

Detailed Description Text (84):

Because the bandwidth obligations are equal for each partition in FIG. 12, each preferably receives an equal number of instruction cycles. However, for partitioning schemes in which the bandwidth obligations differ among the partitions, then the instruction cycles may be apportioned in accordance with the bandwidth requirements. FIG. 14 illustrates an exemplary timing diagram for allocating instruction cycles for a partitioned heap in an interleaved and pipelined manner in accordance with the present invention. In this example, assume that the heap 700 (FIG. 7) is partitioned to provide three OC-48 channels (designated in FIG. 14 as Partitions 1-3) and four OC-12 channels designated in FIG. 14 as Partitions 4-7). This gives a total of seven channels with a combined bandwidth that is equivalent to one OC-192 channel. However, each OC-48 channel carries essentially four times the data traffic as each OC-12 channel. Thus, each heap partition that supports an OC-48 channel preferably receive four times the number of instruction cycles as each heap partition that supports an OC-12 channel.

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)**End of Result Set**

Generate Collection

Print

L10: Entry 5 of 5

File: USPT

Aug 17, 1999

DOCUMENT-IDENTIFIER: US 5938309 A

TITLE: Bit-rate transparent WDM optical communication system with remodulators

Detailed Description Text (21):

To recreate four OC-12 optical signals from the OC-48 .lambda..sub.1, optical channel, the electrical signal from receiver 130 is routed to 1:4 electrical demultiplexer 140. Demultiplexer 140 creates four OC-12 signals from the interleaved time slots of the OC-48 signal. Similarly, the OC-192 optical signal is created by combining the four OC-48 optical channels, .lambda..sub.2 - .lambda..sub.5, in electrical multiplexer 150, placing a portion of each of the OC-48 signals into the appropriate time slot as in the original OC-192 signal.

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)